20/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

06021105 E.I. No: EIP02126889975
Title: A technique for high bandwidth and deterministic low latency load/store accesses to multiple cache banks
Author: Neefs, Henk; Vandierendonck, Hans; De Bosschere, Koen
Corporate Source: Dept. of Electronics and Info. Syst. University of Gent, 9000 Gent, Belgium
Conference Title: Sixth International Symposium on High-Performance Computer Architecture
Conference Location: Toulouse, France Conference Date: 20000108-20000112
Sponsor: IEEE Computer Society; Technical Committee on Computer Architecture
E.I. Conference No.: 59047
Source: IEEE High-Performance Computer Architecture Symposium Proceedings 2000. p 313-324
Publication Year: 2000
CODEN: 85QSAT
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical)
Journal Announcement: 0203W4
Abstract: One of the problems in future processors will be the resource conflicts caused by several load/store units competing to access the same cache bank. The traditional approach for handling this case is by introducing buffers combined with a cross-bar. This approach suffers from (i) the non-deterministic latency of a load/store and (ii) the extra latency caused by the cross-bar and the buffer management. A deterministic latency is of the utmost importance for the forwarding mechanism of out-of-order processors because it enables back-to-back operation of instructions. We propose a technique by which we eliminate the buffers and crossbars from the critical path of the load/store execution. This results in both, a low and a deterministic latency. Our solution consists of **predicting** which bank is to be accessed. Only in the case of a wrong **prediction** a penalty results. 13 Refs.
Descriptors: **Parallel** processing systems; Interconnection networks; Cache memory; Bandwidth; Critical path analysis; Algorithms
Identifiers: **Multiple cache** banks; Deterministic latency; Instruction **level** parallelism
Classification Codes:
722.4 (Digital Computers & Systems); 722.1 (Data Storage, Equipment & Techniques); 921.6 (Numerical Methods)
722 (Computer Hardware); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

20/5/2 (Item 2 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

05840919 E.I. No: EIP01266555123
Title: Optimizations enabled by a decoupled front-end architecture
Author: Reinman, G.; Calder, B.; Austin, T.
Corporate Source: Dept. of Comp. Science and Eng. University of California, San Diego, La Jolla, CA 92093, United States
Source: IEEE Transactions on Computers v 50 n 4 April 2001 2001. p 338-355
Publication Year: 2001
CODEN: ITCOB4 ISSN: 0018-9340
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications)
Journal Announcement: 0106W5
Abstract: In the pursuit of instruction-level parallelism, significant demands are placed on a processor's instruction delivery mechanism. Delivering the performance necessary to meet future processor execution targets requires that the performance of the instruction delivery

mechanism scale with the execution core. Attaining these targets is a challenging task due to I-cache misses, branch mispredictions, and taken branches in the instruction stream. To counter these challenges, we present a fetch architecture that decouples the branch **predictor** from the instruction fetch unit. A Fetch Target Queue (FTQ) is inserted between the branch **predictor** and instruction cache. This allows the branch **predictor** to run far in advance of the address currently being fetched by the cache. The decoupling enables a number of architecture optimizations, including multilevel branch **predictor** design, fetch-directed instruction prefetching, and easier pipelining of the instruction **cache** . For the **multilevel predictor** , we show that it performs better than single-level **predictor** , even when ignoring the effects of cycle-timing issues. We also examine the performance of fetch-directed instruction prefetching using a multilevel branch **predictor** and show that an average 19 percent speedup is achieved. In addition, we examine pipelining the instruction cache to achieve a faster cycle time for the processor pipeline and show that pipelining provides an average 27 percent speedup over not pipelining the instruction cache for the programs examined. 43 Refs.

Descriptors: **Parallel** processing systems; Optimization; Cache memory; Pipeline processing systems; Computer simulation; C (programming language); Program compilers

Identifiers: Fetch target queue; Decoupled front end architecture; Instruction prefetching; Branch **prediction**

Classification Codes:

723.1.1 (Computer Programming Languages)

722.4 (Digital Computers & Systems); 921.5 (Optimization Techniques); 722.1 (Data Storage, Equipment & Techniques); 723.5 (Computer Applications); 723.1 (Computer Programming)

722 (Computer Hardware); 921 (Applied Mathematics); 723 (Computer Software, Data Handling & Applications)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

20/5/3 (Item 3 from file: 8)

05113960 E.I. No: EIP98094368466

Title: Proceedings of the 1998 25th Annual International Symposium on Computer Architecture

Author: Anon (Ed.)

Conference Title: Proceedings of the 1998 25th Annual International Symposium on Computer Architecture

Conference Location: Barcelona, Spain Conference Date: 19980627-19980701

Sponsor: IEEE; ACM SIGARCH

E.I. Conference No.: 48907

Source: Conference Proceedings - Annual International Symposium on Computer Architecture, ISCA 1998. IEEE Comp Soc, Los Alamitos, CA, USA. 391p

Publication Year: 1998

CODEN: CPAAEV ISSN: 0884-7495

Language: English

Document Type: CP; (Conference Proceedings) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9811W1

Abstract: The proceedings contains 33 papers. Topics discussed include machine measurement, program behavior, graphics, speculation, **prediction** techniques, memory management, predication and multipath execution, processor microarchitecture, **parallel** machines and caches and memory systems.

Descriptors: Computer architecture; Data storage equipment; Computer software; Computer graphics; Computer workstations; Computer simulation; Three dimensional computer graphics; **Parallel** processing systems; Buffer storage; Algorithms

Identifiers: Database workload performance; Data compression algorithms; **Multi level** texture **caching** ; Declustered disk array architecture; Speculation control; Memory management; Threaded multiple path execution;

Code partitioning; Sum addressed memory; EiRev
Classification Codes:
722.1 (Data Storage, Equipment & Techniques); 723.1 (Computer Programming); 723.5 (Computer Applications); 722.4 (Digital Computers & Systems); 921.6 (Numerical Methods)
723 (Computer Software); 722 (Computer Hardware); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

20/5/4 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01902199 ORDER NO: AADAA-I3060016
Clock network and phase-locked loop power estimation and experimentation
Author: Duarte, David Enrique
Degree: Ph.D.
Year: 2002
Corporate Source/Institution: The Pennsylvania State University (0176)
Adviser: Mary Jane Irwin
Source: VOLUME 63/07-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 3403. 138 PAGES
Descriptors: ENGINEERING, ELECTRONICS AND ELECTRICAL ; COMPUTER SCIENCE
Descriptor Codes: 0544; 0984
ISBN: 0-493-75501-2

The clock distribution network and the generation circuitry are critical components of current **synchronous** digital systems and are known to consume more than a quarter of the power budget of existing microprocessors. A high-level clock energy model that captures both the dynamic and leakage power components is formulated. The validation results show an average deviation within 506 of circuit-level simulations.

Further, Phased Locked Loops (PLLs), which have been generally used in clock generation, are also crucial for the implementation of Dynamic Voltage Scaling (DVS) mechanisms employed in emerging power conscious processor designs. In order to devise architectural and compiler driven optimizations that exploit the dynamic frequency voltage scaling features, accurate models that capture the performance and power characteristics of the PLL are essential. In addition, many emerging System-on-a-Chip (SOC) designs use multiple PLLs on the same die making it important to estimate the contribution of the PLL to the overall system power. A PLL energy and timing model that accurately estimates the power consumption during both lock and lock-acquisition states is also formulated. The applicability of PLLs as voltage regulators in support of leakage reduction by supply gating is briefly discussed.

The complete clock energy model is incorporated into a cycle-accurate energy simulator for an embedded architecture. This framework is used to study and quantify the influence on clock energy of several architectural-level decisions and their relative impact on the overall system power. These design choices include various **cache** architectures and clock gating at **different levels** (top- **level** distribution network functional unit and gate level). From the software perspective, the influence on clock energy of power-aware memory-oriented compiler optimizations is assessed.

Finally, the model is used to **predict** the role that the clock will have in the total power budget of future designs while carefully capturing the impact of technology scaling. It is shown that as long as leakage power is kept under control, clock power will remain a significant contributor to the total system power.

20/5/5 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01413066 ORDER NO: AADAA-I9514500
RECONFIGURABLE STAGE BUFFER DESIGN FOR MULTILEVEL CACHE **(CACHE)**

Author: CAI, ZHONG-NING
Degree: PH.D.
Year: 1994
Corporate Source/Institution: UNIVERSITY OF MARYLAND (0117)
Chairman: ROBERT W. NEWCOMB
Source: VOLUME 56/01-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 412. 175 PAGES
Descriptors: ENGINEERING, ELECTRONICS AND ELECTRICAL; COMPUTER SCIENCE
Descriptor Codes: 0544; 0984

This dissertation introduces a reconfigurable stage buffer, a buffer with a variable size and **parallel** ports, that holds stored data temporarily while the central processing unit writes data to its **multilevel cache** . A corresponding analytic model is given which leads to a reconfigurable stage buffer design theory.

Because new generation RISC (Reduced Instruction Set Computing) architectures permit memory accesses without preserving program order in general, the reconfigurable stage buffer uses the CPU's store operation reordering, **synchronization** instructions and its reconfigurability to maximize the store parallelism. Therefore, the reconfigurable stage buffer is able to overcome the problem of difficulty in effectively accommodating the data traffic from the CPU in varying workload environments when using a conventional, fixed configuration, stage buffer.

The reconfigurable stage buffer consists of several independent **parallel** buffers which can be implemented either **concurrently** or sequentially depending on upcoming workload patterns. There are software-controlled signals associated with the reconfigurable stage buffer. According to the CPU's **synchronization** instructions, data ownership, data modification status, and access atomicity, the reconfigurable stage buffer decides whether or not to change its configuration. The reconfigurable stage buffer allows a great store implementation flexibility and improves performance while preserving just enough order that desired program behavior can be guaranteed.

A theory supporting the proposed design is presented in which an analytic model is introduced that allows for convenient design and analysis of the reconfigurable stage buffer. The model **predicts** the average traffic versus the number of independent **parallel** buffers and the size of the independent **parallel** buffers. The model also indicates the impact of the cache architecture on the reconfigurable stage buffer design. The **prediction** indicates the potential traffic congestion in the reconfigurable stage buffer and describes how to avoid traffic congestion. By using the analytic model, we can roughly evaluate reconfigurable stage buffer performance and robustness within a short period of time and with a low cost.

An example, the PowerPC, illustrates practical use of the reconfigurable stage buffer architecture and provides a corresponding logic design. The reconfigurable stage buffer logic design includes the details of Tag and Status Interfaces, Data and Address Interfaces, and Control Logic and Buffer Register Arrangement. It is shown that the proposed reconfigurable stage buffer reduces the number of CPU wait states and alleviates store data congestion.

20/5/6 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

7797616 INSPEC Abstract Number: B2004-01-1265D-041, C2004-01-5320G-027
Title: A noise tolerant cache design to reduce gate and sub-threshold leakage in the nanometer regime
Author(s): Agarwal, A.; Roy, K.
Author Affiliation: Sch. of Electr. & Comput. Eng., Purdue Univ., West Lafayette, IN, USA
Conference Title: ISLPED'03. Proceedings of the 2003 International Symposium on Low Power Electronics and Design (IEEE Cat. No.03TH8713) p. 18-21
Publisher: ACM, New York, NY, USA
Publication Date: 2003 Country of Publication: USA xiv+488 pp.

ISBN: 1 58113 682 X Material Identity Number: XX-2003-00943
U.S. Copyright Clearance Center Code: 1-58113-682-X/03/0008$5.00
Conference Title: ISLPED International Symposium on Low-Power Electronics and Design
Conference Sponsor: ACM SIGDA; IEEE Circuits & Syst. Soc.; IEEE Solid-State Circuits Soc.; IEEE Electron Devices Soc
Conference Date: 25-27 Aug. 2003 Conference Location: Seoul, South Korea
Medium: Also availabe on CD-ROM in PDF format
Language: English Document Type: Conference Paper (PA)
Treatment: New Developments (N); Practical (P); Theoretical (T)
Abstract: Scaling devices while maintaining reasonable short channel immunity requires gate oxide thickness of less than 20 AA for CMOS devices beyond the 70 nm technology node. Low oxide thickness gives rise to considerable direct tunneling current (gate leakage). Power dissipation in large caches is dominated by the gate and sub-threshold leakage current. This paper proposes a novel cache that has high noise immunity with improved leakage power. For every bank of SRAM cells, this technique requires an extra diode in **parallel** with a gated-ground transistor connected between the source of NMOS transistors and ground in SRAM cells. The row decoder itself can be used to control the extra gated-ground transistor. Our simulation results on a 70 nm process (Berkeley **Predictive** Technology Model augmented with our gate leakage model) show 39.2% reduction in consumed energy (leakage plus dynamic) in L1 cache and 59.4% reduction in L2 cache energy with less than 2.5% impact on execution time. The technique is applicable to data and instruction **caches** as well as **different levels** of **cache** hierarchy such as the L1, L2, or L3 caches. (15 Refs)
Subfile: B C
Descriptors: cache storage; circuit simulation; CMOS memory circuits; integrated circuit design; integrated circuit modelling; integrated circuit noise; leakage currents; nanoelectronics; semiconductor diodes; SRAM chips
Identifiers: noise tolerant cache design; gate leakage; sub-threshold leakage; nanometer regime; device scaling; short channel immunity; gate oxide thickness; CMOS technology node; oxide thickness; direct tunneling current; power dissipation; noise immunity; SRAM cell ground; **parallel** diode/gated-ground transistor; NMOS transistor source; row decoder; Berkeley **Predictive** Technology Model; gate leakage model; execution time; instruction caches; data caches; cache hierarchy levels; 20 A; 70 nm
Class Codes: B1265D (Memory circuits); B1265A (Digital circuit design, modelling and testing); B2570D (CMOS integrated circuits); B2570A (Semiconductor integrated circuit design, layout, modelling and testing); B2550N (Nanometre-scale semiconductor fabrication technology); B1130B (Computer-aided circuit analysis and design); C5320G (Semiconductor storage) ; C7410D (Electronic engineering computing); C6120 (File organisation)
Numerical Indexing: size 2.0E-09 m; size 7.0E-08 m

20/5/7 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC

7460544 INSPEC Abstract Number: C2003-01-7430-002
Title: Performance prediction for random write reductions: a case study in modeling shared memory programs
Author(s): Ruoming Jin; Agrawal, G.
Author Affiliation: Dept. of Comput. & Inf. Sci., Ohio State Univ., Columbus, OH, USA
Journal: Performance Evaluation Review Conference Title: Perform. Eval. Rev. (USA) vol.30, no.1 p.117-28
Publisher: ACM,
Publication Date: June 2002 Country of Publication: USA
CODEN: PEREDN ISSN: 0163-5999
SICI: 0163-5999(200206)30:1L.117:PPRW;1-7
Material Identity Number: P301-2002-003
Conference Title: SIGMETRICS '02: International Conference on Measurement and Modeling of Computer Systems

Conference Date: 15-19 June 2002 Conference Location: Los Angeles, CA, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: In this paper, we revisit the problem of performance **prediction** on shared memory **parallel** machines, motivated by the need for selecting parallelization strategy for random write reductions. Such reductions frequently arise in data mining algorithms. In our previous work, we have developed a number of techniques for parallelizing this class of reductions. Our previous work has shown that each of the three techniques, full replication, optimized full locking, and cache-sensitive, can outperform others depending upon problem, dataset, and machine parameters. Therefore, an important question is, "Can we **predict** the performance of these techniques for a given problem, dataset, and machine?". This paper addresses this question by developing an analytical performance model that captures a **two - level cache** , coherence **cache** misses, TLB misses, locking overheads, and contention for memory. Analytical model is combined with results from micro-benchmarking to **predict** performance on real machines. We have validated our model on two different SMP machines. Our results show that our model effectively captures the impact of memory hierarchy (**two - level cache** and TLB) as well as the factors that limit parallelism (contention for locks, memory contention, and coherence cache misses). The difference between **predicted** and measured performance is within 20% in almost all cases. Moreover, the model is quite accurate in **predicting** the relative performance of the three parallelization techniques. (22 Refs)
Subfile: C
Descriptors: performance evaluation; shared memory systems; virtual machines
Identifiers: performance **prediction** ; shared memory **parallel** machines; parallelization strategy; random write reductions; data mining algorithms; optimized full locking techniques; full replication techniques; cache-sensitive techniques; analytical performance model; **two - level cache** ; coherence cache misses; TLB misses; locking overheads; micro-benchmarking; SMP machines; memory hierarchy; memory contention
Class Codes: C7430 (Computer engineering); C5440 (Multiprocessing systems); C5220P (Parallel architecture); C5470 (Performance evaluation and testing)

20/5/8 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC

7394807 INSPEC Abstract Number: C2002-11-5220P-008
Title: Cached two - level **adaptive branch** predictors **with** multiple **stages**
Author(s): Egan, C.; Steven, G.; Vintan, L.
Author Affiliation: Hertfordshire Univ., Hatfield, UK
Conference Title: Trends in Network and Pervasive Computing - ARCS 2002. International Conference on Architecture of Computing Systems. Proceedings (Lecture Notes in Computer Science Vol.2299) p.179-91
Editor(s): Schmeck, H.; Ungerer, T.; Wolf, L.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 2002 Country of Publication: Germany xiv+286 pp.
ISBN: 3 540 43409 7 Material Identity Number: XX-2002-01085
Conference Title: Trends in Network and Pervasive Computing - ARCS 2002. International Conference on Architecture of Computing Systems. Proceedings
Conference Date: 8-12 April 2002 Conference Location: Karlsruhe, Germany
Language: English Document Type: Conference Paper (PA)
Treatment: Applications (A); Practical (P)
Abstract: In this paper, we quantify the performance of a novel family of multi-stage two-level adaptive branch **predictors** . In each two-level **predictor** , the PHT of a conventional two-level adaptive branch **predictor** is replaced by a **prediction** cache. Unlike a PHT, a **prediction** cache

saves only relevant branch **prediction** information. Furthermore, **predictions** are never based on uninitialised entries and interference between branches is eliminated. In the case of a **prediction** cache miss in the first stage, our two-stage **predictors** use a default two-bit **prediction** counter stored in a second stage. We demonstrate that a two-stage cached **predictor** is more accurate than a conventional two-level **predictor** and quantify the crucial contribution made by the second **prediction** stage in achieving this high accuracy. We then extend our cached **predictor** by adding a third stage and demonstrate that a three-stage cached **predictor** further improves the accuracy of cached **predictors** . (16 Refs)
Subfile: C
Descriptors: cache storage; **parallel** architectures; performance evaluation
Identifiers: **cached two - level** adaptive branch **predictors** ; multiple stages; performance evaluation; **prediction** cache; default two-bit **prediction** counter
Class Codes: C5220P (Parallel architecture); C5470 (Performance evaluation and testing); C6120 (File organisation)

20/5/9 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC

7081944 INSPEC Abstract Number: C2001-12-5220P-014
Title: Applying caching **to** two - level **adaptive branch** prediction
Author(s): Egan, C.; Steven, G.B.; Won Shim; Vintan, L.
Author Affiliation: Hertfordshire Univ., Hatfield, UK
Conference Title: Proceedings Euromicro Symposium on Digital Systems Design p.186-93
Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA
Publication Date: 2001 Country of Publication: USA xii+476 pp.
ISBN: 0 7695 1239 9 Material Identity Number: XX-2001-02016
U.S. Copyright Clearance Center Code: 0 7695 1239 9/2001/$10.00
Conference Title: Proceedings Euromicro Symposium on Digital Systems Design
Conference Sponsor: Euromicro
Conference Date: 4-6 Sept. 2001 Conference Location: Warsaw, Poland
Language: English Document Type: Conference Paper (PA)
Treatment: Applications (A); Practical (P)
Abstract: During the 1990s Two-level Adaptive Branch **Predictors** were developed to meet the requirement for accurate branch **prediction** in high-performance superscale processors. However, while two-level adaptive **predictors** achieve very high **prediction** rates, they tend to be very costly. In particular, the size of the second level Pattern **History** Table (PHT) increases exponentially as a function of **history** register length. Furthermore, many of the **prediction** counters in a PHT are never used; **predictions** are frequently generated from non-initialised counters and several branches may update the same counter, resulting in interference between branch **predictions** . In this paper, we propose a **Cached** Correlated **Two - Level** Branch **Predictor** in which the PHT is replaced by a **Prediction** Cache. Unlike a PHT, the **Prediction** Cache saves only relevant branch **prediction** information. Furthermore, **predictions** are never based on uninitialised entries and interference between branches is eliminated. We simulate three versions of our Cached Correlated Branch **Predictors** . The first **predictor** is based on global branch **history** information while the second is based on local branch **history** information. The third **predictor** exploits the ability of cached **predictors** to combine both global and local **history** information in a single **predictor** . We demonstrate that our **predictors** deliver higher accuracy than conventional **predictors** at a significantly lower cost. (13 Refs)
Subfile: C
Descriptors: cache storage; **parallel** architectures; program compilers
Identifiers: caching; two-level adaptive branch **prediction** ; high-performance superscale processors; cached correlated branch

predictors ; local **history** information
Class Codes: C5220P (Parallel architecture); C6150C (Compilers, interpreters and other processors); C6120 (File organisation)

20/5/10 (Item 5 from file: 2)
DIALOG(R)File 2:INSPEC

6376755 INSPEC Abstract Number: C1999-11-6120-022
Title: Efficient analytical modelling of multi - level **set-associative** caches
Author(s): Harper, J.S.; Kerbyson, D.J.; Nudd, G.R.
Author Affiliation: High Performance Syst. Group, Warwick Univ., Coventry, UK
Conference Title: High-Performance Computing and Networking. 7th International Conference, HPCN Europe 1999. Proceedings p.473-82
Editor(s): Sloot, P.; Bubak, M.; Hoekstra, A.; Hertzberger, B.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 1999 Country of Publication: Germany xxiii+1318 pp.
ISBN: 3 540 65821 1 Material Identity Number: XX-1999-02493
Conference Title: High-Performance Computing and Networking. 7th International Conference, HPCN Europe 1999. Proceedings
Conference Date: 12-14 April 1999 Conference Location: Amsterdam, Netherlands
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: The time a program takes to execute is significantly affected by the efficiency with which it utilises cache memory. Moreover the cache miss behaviour of a program is highly unstable, in that small changes to input parameters can cause large changes in the number of misses. In this paper we describe novel analytical methods of **predicting** the cache miss ratio of numerical programs, for sequential hierarchies of set-associative caches. The methods are demonstrated to be applicable to most loop nests. They are also shown to be highly accurate, yet able to be evaluated orders of magnitude faster than a comparable simulation. (12 Refs)
Subfile: C
Descriptors: cache storage; **parallel** programming; virtual machines
Identifiers: efficient analytical modelling; **multi - level** set-associative **caches** ; program execution time; cache memory utilisation; cache miss ratio **prediction** ; numerical programs; sequential hierarchies; loop nest; simulation
Class Codes: C6120 (File organisation); C6150N (Distributed systems software)

20/5/11 (Item 6 from file: 2)
DIALOG(R)File 2:INSPEC

5835514 INSPEC Abstract Number: C9803-5220P-057
Title: Instruction cache **prefetching using** multilevel **branch** prediction
Author(s): Veidenbaum, A.V.
Author Affiliation: Dept. of Electr. Eng. & Comput. Sci., Illinois Univ., Chicago, IL, USA
Conference Title: High Performance Computing. International Symposium, ISHPC '97. Proceedings p.51-70
Editor(s): Polychronopoulos, C.; Joe, K.; Araki, K.; Amamiya, M.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 1997 Country of Publication: Germany xii+416 pp.
ISBN: 3 540 63766 4 Material Identity Number: XX97-02735
Conference Title: High Performance Computing. International Symposium, ISHPC'97. Proceedings
Conference Date: 4-6 Nov. 1997 Conference Location: Fukuoka, Japan
Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)
Abstract: Presents an instruction cache prefetching mechanism that is capable of prefetching **past** branches in multiple-issue processors. At high clock rates, such processors often use small instruction caches which have significant miss rates. Prefetching from a secondary cache can hide the instruction cache miss penalties, but only if initiated sufficiently far ahead of the current program counter (PC). Existing instruction cache prefetching methods are strictly sequential and cannot do that, due to their inability to prefetch **past** branches. By keeping branch **history** and branch target addresses, we **predict** a future PC several branches **past** the current branch. We describe a possible prefetching architecture and evaluate its accuracy, the impact of the instruction prefetching on performance, and its interaction with sequential prefetching. For a 4-issue processor and a cache architecture patterned after the DEC Alpha-21164, we show that our prefetching unit can be more effective than sequential prefetching. The two types of prefetching eliminate different types of misses and can thus be effectively combined to achieve better performance. (28 Refs)
Subfile: C
Descriptors: cache storage; memory architecture; **parallel** architectures ; performance evaluation
Identifiers: instruction cache prefetching mechanism; multilevel branch **prediction** ; multiple-issue processors; clock rate; miss rates; secondary cache; instruction cache miss penalty hiding; program counter **prediction** ; **past** branches; branch **history** ; branch target addresses; prefetching architecture; accuracy; performance; sequential prefetching; 4-issue processor; cache architecture; DEC Alpha-21164; instruction-level parallelism
Class Codes: C5220P (Parallel architecture); C5470 (Performance evaluation and testing)

20/5/12 (Item 7 from file: 2)
DIALOG(R)File 2:INSPEC

4754663 INSPEC Abstract Number: C9410-6120-018
Title: Scalable temporally predictable **memory structures**
Author(s): Moore, S.W.
Author Affiliation: Comput. Lab., Cambridge Univ., UK
p.99-103
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1994 Country of Publication: USA xi+167 pp.
ISBN: 0 8186 6375 8
U.S. Copyright Clearance Center Code: 0 8186 6375 8/94/$04.00
Conference Title: Proceedings of 2nd IEEE Workshop on Real-Time Applications
Conference Sponsor: IEEE Comput. Soc. Tech. Committee on Real-Time Syst
Conference Date: 21-22 July 1994 Conference Location: Washington, DC, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Faster processors are used to tackle larger problems which typically require a larger memory. Unfortunately this prohibits memory access latency from scaling with processor speed, Consequently, **multiple levels** of **caching** are employed which utilise temporal and spatial locality of reference to bridge the performance gap. However, cache performance is difficult to **predict** which is problematic for hard real-time systems. A tree memory structure, whose access frequency, rather than latency, can scale with processor speed, is proposed, together with a scalable memory module base virtual addressing mechanism and page based memory protection using capabilities. It is concluded that a multi-threaded processor would be desirable to utilise the **concurrency** of hard real-time applications to tolerate the latency of the memory tree. (8 Refs)
Subfile: C
Descriptors: buffer storage; multiprocessing systems; performance evaluation; real-time systems; virtual storage

Identifiers: scalable temporally **predictable** memory structures; memory access latency; processor speed; caching; cache performance; hard real-time systems; tree memory structure; scalable memory module; virtual addressing mechanism; page based memory protection; multithreaded processor; hard real-time applications; **concurrency** ; memory tree
Class Codes: C6120 (File organisation); C5320G (Semiconductor storage); C5470 (Performance evaluation and testing); C5440 (Multiprocessor systems and techniques)

20/5/13 (Item 1 from file: 34)
DIALOG(R)File 34:SciSearch(R) Cited Ref Sci

02144314 Genuine Article#: KE678 Number of References: 3
Title: DESIGN OF THE IBM ENTERPRISE SYSTEM/9000 HIGH-END PROCESSOR
Author(s): LIPTAY JS
Corporate Source: IBM CORP,ENTERPRISE SYST,POB 950/POUGHKEEPSIE//NY/12602
Journal: IBM JOURNAL OF RESEARCH AND DEVELOPMENT, 1992, V36, N4 (JUL), P 713-731
ISSN: 0018-8646
Language: ENGLISH Document Type: ARTICLE
Geographic Location: USA
Subfile: SciSearch; CC PHYS--Current Contents, Physical, Chemical & Earth Sciences; CC ENGI--Current Contents, Engineering, Technology & Applied Sciences
Journal Subject Category: COMPUTER APPLICATIONS & CYBERNETICS
Abstract: The "high-end" water-cooled processors in the IBM Enterprise System/9000(TM) product family use a CPU organization and cache structure which depart significantly from previous designs. The CPU organization includes multiple execution elements which execute instructions out of sequence, and uses a new virtual register management algorithm to control them. It also contains a branch **history** table to remember recent branches and their target addresses so that instruction fetching and decoding can be directed more accurately. These models also use a **two - level cache** structure which provides a **level** 1 **cache** associated with each processor and a level 2 cache associated with central storage. The level 1 cache uses a store-through organization, and is split into two separate caches, one used for instruction fetching and the other for operand references. The level 2 cache uses a store-in method to handle stores.
Research Fronts: 91-1020 001 (DEBUGGING **PARALLEL** PROGRAMS; SYSTOLIC ARCHITECTURE; MULTIPROCESSOR SYSTEMS; MAPPING NESTED LOOPS; SCALAR COMPILATION TECHNIQUES; VLIW COMPUTERS)
Cited References:
LIPTAY JS, 1968, V7, P15, IBM SYST J
TOMASULO RM, 1967, V11, P25, IBM J RES DEV
TUCKER SG, 1986, V25, P4, IBM SYST J

20/5/14 (Item 1 from file: 95)
DIALOG(R)File 95:TEME-Technology & Management

00761357 E94024981020
A multi - level **hierarchical** cache **coherence protocol for multiprocessors**
(Ein Coherence Protokoll fuer Multiprozessoren mit hierarchisch strukturiertem Mehrebenen-Cache)
Anderson, C; Baer, J-L
Univ. of Washington, Seattle, USA
7th Int. Parallel Processing Symp. Proc., Newport Beach, USA, Apr 13-16, 19931993
Document type: Conference paper Language: English
Record type: Abstract
ISBN: 0-8186-3442-1

ABSTRACT:

In order to meet the computational needs of the next decade, shared-memory processors must be scalable. Though single shared-bus architectures have been successful in the **past** , lack of bus bandwidth restricts the number of processors that can be effectively put on a single bus machine. One architecture that has been proposed to solve the limited bandwidth problem consists of processors connected via a tree hierarchy of buses. In this paper, the authors present a tool to study a hierarchical bus based shared-memory system. The authors highlight the main features of a hierarchical cache coherence protocol and give some preliminary performance results obtained via an instruction level simulator.

DESCRIPTORS: PROGRAM SYSTEM; SYSTEM DESCRIPTION; COMMAND LANGUAGES; PROGRAM INSTRUCTION; MASSIVELY **PARALLEL** MACHINES; COMPUTER ARCHITECTURE; **PARALLEL** PROCESSORS; INTERCONNECTED OPERATION; **PARALLEL** PROCESSING; DATA BUS; BUS STRUCTURE; DATA SIGNALLING RATE; SOFTWARE TOOLS; MEMORY MANAGEMENT ; HIERARCHY; SIMULATION PROGRAMS; MODEL SIMULATION; PERFORMANCE EVALUATION; COMPUTERIZED SIMULATION; COMMAND STRUCTURE; CACHE MEMORIES
IDENTIFIERS: Coherence Protokoll; Mehrebenencache